Scale-rule selection of affordable neural network for chaotic time series learning

Uwate, Y ; Nishio, Y ; Stoop, R

DOI: https://doi.org/10.1109/ECCTD.2007.4529720

Scale-Rule Selection of Affordable Neural Network for Chaotic Time Series Learning

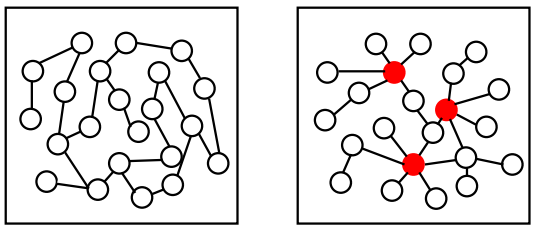
Yoko Uwate and Yoshifumi Nishio
Dept. of Electrical and Electronic Engineering,
Tokushima Univ., 2-1 Minami-Josanjima, Tokushima, Japan
Email: {uwate, nishio}@ee.tokushima-u.ac.jp

Ruedi Stoop
Institute of Neuroinfomatics, University / ETH Zurich
Winterthurerstrasse 190, 8057 Zurich, Switzerland
Email: ruedi@ini.phys.ethz.ch

***Abstract*— Scale-free networks are an important class of complex networks since many "real-world networks" fall into this category. In our contribution we investigate the influence of this property on the performance of an affordable neural network. By means of computer simulations, we confirm that affordable neural networks, when the affordable neurons are chosen in a scale-free manner, perform significantly better compared to random selection.**

I. Introduction

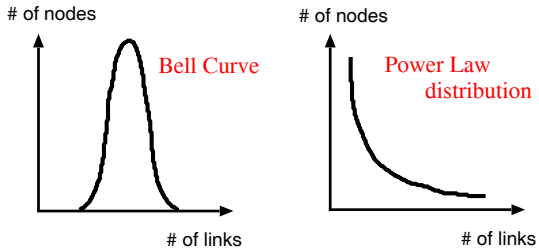
Since the scale-free networks were discovered by Barabasi et al. [1], studies assessing the influence of this property on the efficiency of networks have been carried out in various fields. One way of how to characterize the difference between random and scale-free networks is by means of the distribution of the number of links ♯ a node has. From Fig. 1, where we contrast the two network types, it is evident that in the scale-free network, although most nodes only have few connections, some nodes (marked in red) act as highly connected hubs. This distinction is captured in a more quantitative way by the distribution of the number of links vs. the number of nodes, as shown in Figure 2. Random networks display a bell-shaped curve, implying that most nodes have the same number of links, and no highly connected nodes (see Fig. 2 (a)). Scale-free networks, in contrast, often have many nodes with a few links only, whereas quite a few hubs exist that have a large number of links. Mathematically, scale-free networks are characterized by power law distributions (Fig. 2 (b)). Because scale-rules emerge in many areas and disciplines of science (e.g. engineering, economics, social sciences and so on), we expect that also in the development of the science of complex networks, they will play an important role.



(a) Random network. (b) Scale-free network.

Fig. 1. Example of random and scale-free network.

In a previous study on artificial neural networks, we proposed a new network structure with affordable neurons in the hidden layer, for efficient BP-learning [2]. We christened this network "Affordable Neural Network." In this network, some extra neurons are inserted into the hidden layer. When the network operates, not all of the neurons present in the hidden layer are updated; the affordable neurons are by-passed. By using different sets of neurons for the update in the hidden layer, the network is able to operate at a high performance. By computer simulations [2], the affordable neural network has been confirmed to achieve an improved performance over conventional networks for BP-learning, in terms of speed of convergence and of learning efficiency. Moreover, we have investigated the performance of the affordable neural network for noise-polluted input data. We found that the affordable neural network is able to generate noise-cleaned outputs, which leads to the conclusion that the affordable neural network has the generalization property. However, we believe that many advantageous characteristics of the affordable neural network are yet to be unveiled, and we also believe that the operation of the affordable neural network embodies important general features of the BP-learning process.



(a) Random network. (b) Scale-free network.

Fig. 2. Comparing random and scale-free distribution.

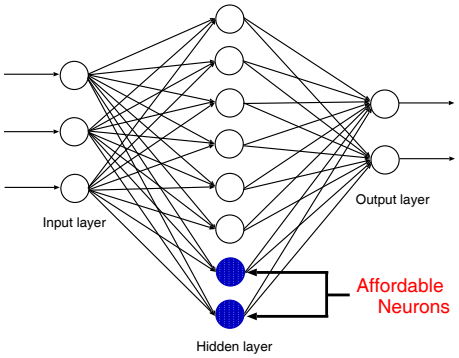
The first step in this programme deals with the performance of the affordable neural network if the affordable neurons are selected by a scale-rule, instead of making a random choice. We will show by computer simulations, that the affordable neural network under scale-rule selection achieves a better performance, compared to random selection.

II. Affordable Neural Network

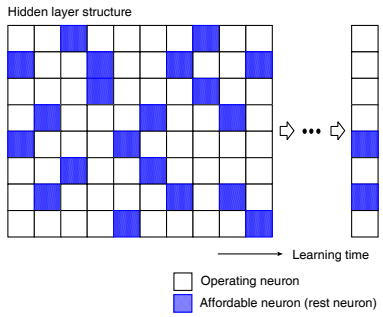
A. Network Model with Affordable Neurons

In Ref. [2], we introduced the affordable neurons to reflect important properties of the brain. During BP-learning, not all

of the neurons in the hidden layer are used at every updating: some of the neurons are selected for the learning and the rest of the neurons are deactivated. The underlying network model and the described mode of operation of the affordable neurons in the hidden layer is sketched in Figs. 3(a) and (b), respectively.



(a) Network model with affordable neurons.



(b) Operation of the affordable neurons in the hidden layer.

Fig. 3. Affordable neural network.

B. Selection of Affordable Neurons

The two models of the selection of the affordable neurons in the hidden layer are implemented as follows.

1) Scale-rule selection: Our scale-rule selection procedure is described in terms of a parameter denoted by a vector S. The dimension s of S equals the number of neurons present in the hidden layer; each component of S corresponds to one single neuron indexed by i. The values of the components are evaluated in each update by

$$S_i = random()/i^2 \tag{1}$$

where $random()$ means the uniform random function producing values from 0.0 to 1.0. This implies that the neuron with the highest index will generally have a small value, whereas the first neuron will – unless the random function states something different – have a larger entry. Note that the values of the entries follow a power-law distribution. Using these values, we select in each update the set of active neurons according to the values of S. From the s neurons in the hidden layer, exactly the k neurons with the smallest entries are chosen as the affordable neurons. Figure 4 illustrates this scale-rule selection of the affordable neurons, where the number of the neurons in the hidden layer is set to be 100 and the number of affordable neurons is 20, 40 and 60, respectively. Our simulations will be based on 100000 updates. In this Figure, the horizontal axis indicates the neuron number, whereas the vertical axis displays the number of times the corresponding neuron was in the set of operating neurons. By this Figure it is confirmed that the operation time decreases gradually with the neuron number of the hidden layer. Furthermore, histograms of the number of neurons that have a given operation time are shown in Fig. 5. The resemblance with the scale-free distribution of Fig. 2 (b) is evident, although, when inspected in details, the distribution is not of a simple power law type.

2) Random selection: For comparison let us also consider random selection of affordable neurons. The results we obtain are shown in Fig. 6. As can be expected, under random selection, the operation time is similar for any neuron. Figure 7 shows again the distribution relationship between the selected time and the number of neurons. This curve clearly reproduces the bell-shaped distribution.

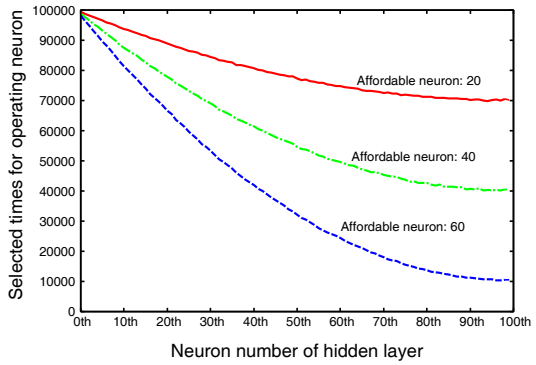
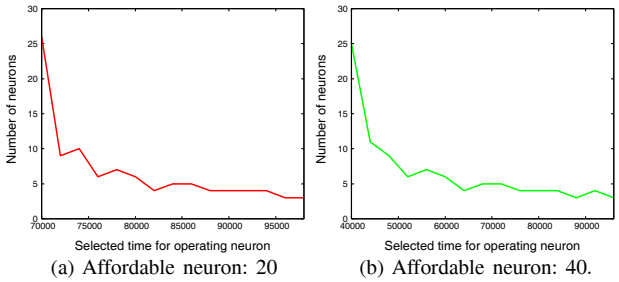
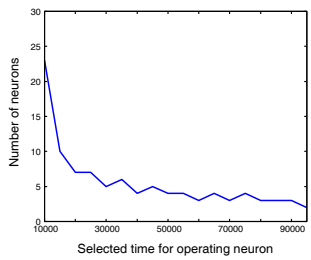


Fig. 4. Scale-rule selection (Hidden: 100).



(a) Affordable neuron: 20 (b) Affordable neuron: 40.



(c) Affordable neuron: 60.

Fig. 5. Distribution of scale-rule selection (Hidden: 100).

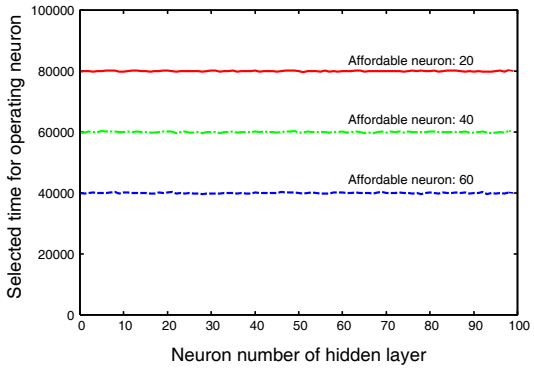
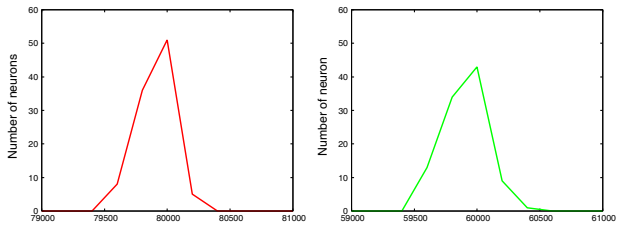
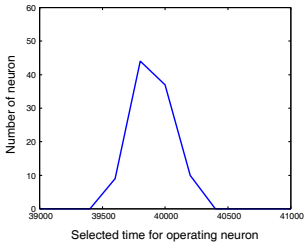


Fig. 6. Random selection (Hidden: 100).



(a) Affordable neuron: 20 (b) Affordable neuron: 40.



(c) Affordable neuron: 60.

Fig. 7. Distribution of random selection (Hidden: 100).

III. BP-Learning Algorithm

BP is the most common learning algorithm for feedforward neural networks, and the effectiveness of BP-learning has been confirmed in pattern recognition, system control, signal processing, and so on [4]-[6]. The standard BP-learning algorithm was introduced in [3]. In this study, we use the batch BP-learning algorithm. The batch BP-learning algorithm can be expressed similarly to the standard BP-learning algorithm, with the difference lying in the timing of the weights. In standard BP, the update is performed after each single data input, whereas for batch BP, the update is performed after all input data has been processed. The total error E of the network is thus defined as

$$E = \sum_{p=1}^{P} E_p = \sum_{p=1}^{P} \left\{ \frac{1}{2} \sum_{i=1}^{N} (t_{pi} - o_{pi})^2) \right\}, \qquad (2)$$

where P is the number of the input data, N is the number of the neurons in the output layer, t_{pi} denotes the value of the desired target data for the pth input data, and o_{pi} denotes the value of the output data for the pth input data. The goal of the learning is to obtain weights between all layers of the network that minimize the total error E. In order to achieve this, the weights are adjusted according to the equation

$$w_{i,j}^{k-1,k}(m+1) = w_{i,j}^{k-1,k}(m) + \sum_{p=1}^{P} \Delta_p w_{i,j}^{k-1,k}(m),$$
$$\Delta_p w_{i,j}^{k-1,k}(m) = -\eta \frac{\partial E_p}{\partial w_{i,j}^{k-1,k}}, \qquad (3)$$

where $w_{i,j}^{k-1,k}$ is the weight between the ith neuron of the $k-1$th layer and the jth neuron of the layer k, m is the learning time, and η is a proportionality factor known as the learning rate. In this study, we also add an inertia term, which changes the second line of Eq.(2) into

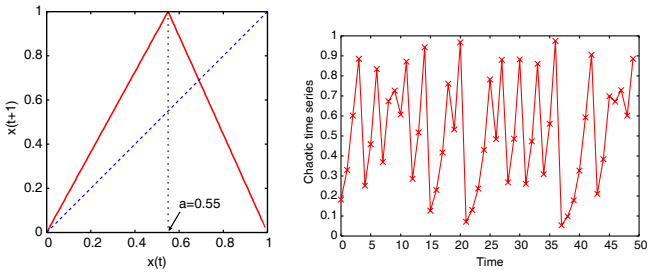
$$\Delta_p w_{i,j}^{k-1,k}(m) = -\eta \frac{\partial E_p}{\partial w_{i,j}^{k-1,k}} + \zeta \Delta_p w_{i,j}^{k-1,k}(m-1), \qquad (4)$$

where ζ denotes the inertia rate.

IV. Simulated Results

For our simulations we want to teach our network to generate typical time series of the skew tent map. To this end, the network is trained – using time series of the tent map – to output, starting from given initial conditions, the same time series as the tent map would have generated.

The skew tent map and an example of time series are shown in Fig. 8. The length of chaotic time series is set to 50 steps; the size of the set of learning patterns is 20. In our approach, this requires the network to have 50 nodes in the input and the output layers. Each data is inputted to each node in the input layer. We carried out BP-learning by using the following parameters. The parameter of the learning rate and the inertia rate are fixed at $\eta = 0.05$ and $\zeta = 0.02$, respectively. The initial values of the weights are chosen between -1.0 and 1.0 at random. The learning time is set to 5000.



(a) Skew tent map. (b) Time series.

Fig. 8. Skew tent map.

A. Learning Ability

First, we investigate the learning efficiency as the average of the total error between the output and the desired target, when the network structure of the hidden layer is changed. The "Average Error E_{ave}" for this learning example is defined by the following equation.

$$E_{ave} = \frac{1}{P} \sum_{p=1}^{P} \left\{ \frac{1}{2} (t_p - o_p)^2 \right\} \qquad (5)$$

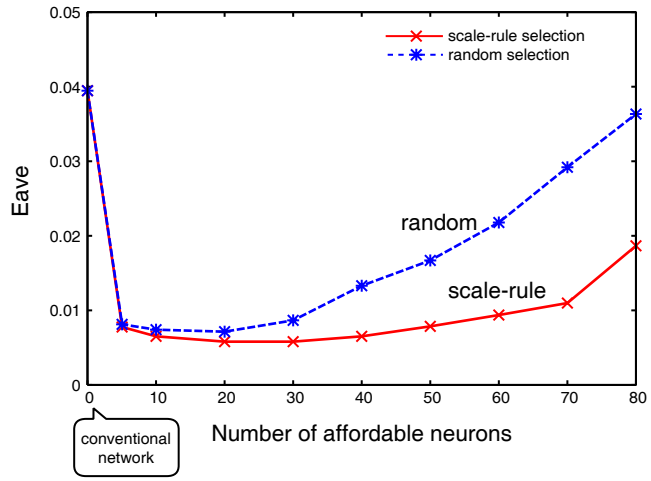


Fig. 9. Learning ability by changing the number of affordable neurons (Number of neurons in hidden layer: 100).

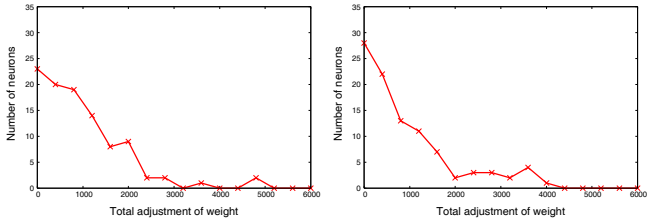
We consider the case that the hidden layer consists of 100 neurons. The number of the affordable neurons is varied from 10 to 70. The results of this simulation are shown in Fig. 9, where the horizontal axes are the number of the affordable neurons and the vertical axes are E_{ave} for the pattern learning. From this Figure, we can confirm that the scale-rule selection method achieves a better performance if compare to the random selection. It is also seen that the difference between the errors of the scale-rule and the random selection networks increases with the number of affordable neurons. Even when the number of affordable neurons becomes large, the scale-rule selection network continues to show good learning ability. From this result, we can conclude that the scale-rule selection method of affordable neurons could play an important role for learning processes, in particular in biological systems.

B. Network Characteristics

In this Section, we compare the characteristics of the affordable neural network under the scale-rule selection to the random selection methods. We pay attention to the total adjustment of the weights of neurons in the hidden layer. The total adjustment of the weights (T_{aw}) is defined as the sum of the absolute values of the adjustments of the weights between input and output. This can be written as

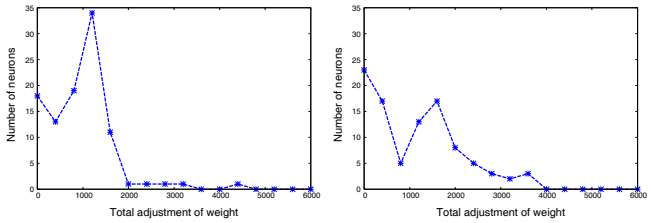
$$T_{aw} = \sum_{r=1}^{R} |\Delta\omega_{i,j}^{1,2} + \Delta\omega_{i,j}^{2,3}|, \qquad (6)$$

where R is the updating time. The results of the relationship between total adjustment of the weight and the number of the neurons obtained in our simulations are shown in Figs. 10 and 11. In the case of scale-rule selection, the number of the neurons decreases gradually, by increasing the total adjustment of the weight (Fig. 10). In contrast, the graph of the random selection has a peak (Fig. 11). In this way, we have confirmed that the operation of neurons in the hidden layer of affordable neural networks with the scale-rule selection follows some scaling rule. We conclude that with the scale-rule characteristics, the performance of affordable neural networks can be boosted considerably.



(a) Affordable neurons: 20. (b) Affordable neurons: 60.

Fig. 10. Network characteristics of scale-rule selections (Hidden:100).



(a) Affordable neurons: 20. (b) Affordable neurons: 60.

Fig. 11. Network characteristics of random selections (Hidden:100).

V. Conclusions

In this study, we investigated the performance of affordable neural networks when the affordable neurons are selected by a scale-rule. By computer simulations, we confirmed that the affordable neural network with scale-rule selection achieves a better performance compared to random selection. Furthermore, we have provided evidence that the operation of neurons in the hidden layer of affordable neural network with scale-rule selection follows a scaling rule. We thus conclude that in the context of affordable neural networks, scale-rule selection of affordable neurons has an important impact on its performance.

References

[1] Barabasi, Albert-Laszlo and A. Reka, "Emergence of Scaling in Random Networks," *Science*, 286:509-512, Oct. 1999.

[2] Y. Uwate and Y. Nishio, "Performance of Affordable Neural Network for Back Propagation Learning," *IEICE Transactions on Fundamentals*, vol. E89-A, no. 9, pp. 2374-2380, Sep. 2006.

[3] D.E. Rumelhart, G.E. Hinton and R.J. Williams, "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing*, vol. 1, MIT Press, MA, pp. 318-362, 1986.

[4] Y. LeCun, B. Boser, J.S. Denker, et al, "Handwritten Digit Recognition with a Back-Propagation Network," *Advances in Neural Information Processing Systems*, vol. 2, MIT Press, MA, 1990.

[5] M. Kawano, Y. Uno, M. Isobe and R. Suzuki, "A Hierarchical Neural Network Model for Voluntary Movement with Application to Robotics," *IEEE Control Systems Magazine*, vol. 8, pp. 8-16, 1988.

[6] T.J. Sejnowski and C.R. Rosenberg, "Parallel Networks that Learn to Pronounce English Text," *Complex Systems*, vol. 1, pp. 145-168, 1987.